

REMARKS

Claims 1-14 are pending.

The final Office Action mailed June 17, 2005 rejected claims 1-14 under 35 U.S.C. § 102(e) as anticipated by *Friesen* (U.S. 6,636,863).

The rejection of the claims is respectfully traversed, as *Friesen* does not disclose the features of the claims. For example, independent claim 1 recites: “loading the class containing the pre-initialized static variable into a **shared, read-only memory**.” Independent claim 5 recites: “loading a class containing a static variable into a **shared, read-only memory**,” and independent claim 8 recites: “loading a class containing a static variable into a **shared, read-only memory**.” This feature is not shown in *Friesen*.

To anticipate, every element and limitation of the claimed invention must be found in a single prior art reference, arranged as in the claim. *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383, 58 USPQ2d 1286, 1291 (Fed. Cir. 2001); *Scripps Clinic & Research Foundation v. Genentech, Inc.*, 927 F.2d 1565, 1576, 18 USPQ2d 1001, 1010 (Fed. Cir. 1991).

Friesen is directed to generating programs associated with Internet web pages that can persist from one web page to another without requiring repeated downloading of the programs from a remote server (col. 1:14-18). Accordingly, *Friesen* discloses a technique where “program code is added to the HTML of each of a plurality of web pages that is loaded and executed by the software engine to generate the object when a first of the web pages is loaded on the computer” (col. 3:15-18). This program is identifiable to the software engine so that when the program is reactivated the object can be used “without reloading the program from the HTML code of the web page” (col. 3:26-27).

The “key” to this technique, as identified by *Friesen* (col. 3:28), is that “a number of variables are defined in the program as static variables **that can be modified during execution of the program**” (col. 3:28-30, emphasis added). *Friesen* states (col. 3:28-51):

The key to the first technique is that a number of variables are defined in the program as static variables that can be modified during execution of the program. Static variables in JAVA have the characteristic that they will be remembered by the software engine even if the program is suspended or rendered inactive by loading of another web page. Thus, in a shopping cart example, if the identities, quantities and prices of selected items are defined in the shopping cart program as static variables, whose values can be changed through selection of items from an associated web page, then the values of these variables will remain stored by the JVM in memory along with the rest of the program code when another web page is loaded. If this new web page then also includes a reference or call to the same applet or program, the JVM will reactivate the shopping cart program along with the values of the static variables as they were left in the last page. The result is that through this repeated reactivation of the program and static variables stored in memory by the JVM, the shopping cart can be made to appear as if it continually persists as a purchaser surfs through a group of shopping site web pages. Further, there is no need for the purchaser's or client's computer or other device to communicate with the shopping site server until they are ready to submit their complete order for processing.

In the “Response to Arguments” section, the Office Action (p. 2) states, “modifying values for static variables in a shopping cart frame or instance … has no effect on the **class** containing the defined static variables.” The Office Action then states (pp. 2-3):

It is inherent the static String is pre-initialized to ‘update.txt’ in said class. It is further inherent and essential to Friesen’s invention that said class is loaded into a shared, read-only memory. Since if the class is not in the shared, read-only memory, itemPrice can be freely modified for any corresponding itemID. Furthermore, Friesen’s invention is directed [*sic*] persisting a shopping cart object from one web page to another. It is inherent that classes (containing static variables) implementing the shopping cart object are persisted in memory (i.e., **read-only**) from one web page to another (i.e., **shared** by web pages), thus enables Friesen to eliminate the need to download the same classes (in order to maintain the shopping cart object) every time the user accesses a different web page).

A single prior art reference anticipates a patent claim if it expressly or inherently describes each and every limitation set forth in the patent claim. *Verdegaal Bros., Inc. v. Union*

Oil Co., 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).). Inherent anticipation requires that the missing descriptive material is “necessarily present,” not merely probably or possibly present, in the prior art. *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citing *Continental Can Co. USA, Inc. v. Monsanto Co.*, 948 F.2d 1264, 1268, 20 USPQ2d 1746, 1749 (Fed. Cir. 1991)). To the extent that the Office Action relies on “common knowledge” to make these assertions regarding the supposed “read-only memory,” Applicants respectfully submit that the APA requires the Patent Office to articulate and place on the record the “common knowledge” used to negate patentability. *In re Zurko*, No. 96-1285 (Fed. Cir., Aug. 2, 2001). *In re Lee*, 277 F.3d 1338, 1344-45, 61 USPQ2d 1430, 1434-35 (Fed. Cir. 2002). Ordinarily, there must be some form of evidence in the record to support an assertion of common knowledge. See *Lee*, 277 F.3d at 1344-45, 61 USPQ2d at 1434-35 (Fed. Cir. 2002); *Zurko*, 258 F.3d at 1386, 59 USPQ2d at 1697 (holding that general conclusions concerning what is “basic knowledge” or “common sense” to one of ordinary skill in the art without specific factual findings and some concrete evidence in the record to support these findings will not support an obviousness rejection).

Applicants further respectfully submit that the supposed “read-only memory” alleged by the Office Action to be inherent in *Friesen* as discussed above is not “necessarily present,” and in fact is not possible, as *Friesen* explicitly discloses that the reactivated shopping cart program will contain static variables that are modified by execution from one activation to the next.

However, independent claims 1, 5, and 8 specifically recite loading into a “shared, read-only memory.” Thus, *Friesen* not only fails to disclose the invention as defined in the claim, *Friesen* actively teaches against the recitation of loading into a “shared, read-only memory.” No discussion of shared, read-only memory can be found in *Friesen*, which is only reasonable, since

Friesen's self-identified "key" technique does not work with shared, read-only memory. Thus, the rejection of claims 1-14 should be withdrawn.

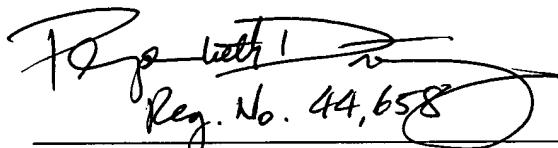
Therefore, the present application overcomes the objections and rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the undersigned attorney at 703-425-8501 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

8/17/05

Date



Margo Livesay, Ph.D.
Attorney/Agent for Applicant(s)
Reg. No. 41,946

10507 Braddock Rd
Suite A
Fairfax, VA 22032
Tel. 703-425-8501
Fax. 703-425-8518